



**HAL**  
open science

## Protection remontante des communications multicast

Mohand Yazid Saidi, Bernard Cousin, Miklos Molnar

► **To cite this version:**

Mohand Yazid Saidi, Bernard Cousin, Miklos Molnar. Protection remontante des communications multicast. MANifestation des JEunes Chercheurs en Sciences et Technologies de l'Information et de la Communication (MajecSTIC), Nov 2006, Lorient, France. hal-00530271

**HAL Id: hal-00530271**

**<https://hal.science/hal-00530271>**

Submitted on 28 Oct 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Protection remontante des communications multicast

**Mohand Yazid SAIDI, Bernard COUSIN et Miklós MOLNÁR**  
Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA)  
Campus Universitaire de Beaulieu  
35042 Rennes, France  
msaidi@irisa.fr, bcousin@irisa.fr, molnar@irisa.fr

**Résumé :** La protection des communications multicast devient de plus en plus importante avec l'avènement et la multiplication des applications multicast temps réel telles que la vidéo à la demande, la vidéo conférence, le jeu en réseau, etc. Ces applications sont sensibles aux ruptures des communications et ont besoin de tourner sur des réseaux robustes et permettant une récupération rapide des pannes. Pour ce faire, différentes techniques de protection multicast proactive ont été développées. Ces techniques permettent de maintenir les communications (ou de diminuer des temps de leur rupture) grâce au pré-calcul et plus souvent à la pré-configuration de chemins de secours utilisables après la détection d'une panne.

Dans cet article, nous présenterons la technique de *protection remontante* qui permet de protéger efficacement les communications multicast par l'utilisation d'une forêt de secours. Cette dernière est construite par l'interconnexion des nœuds de l'arbre primaire en commençant par les nœuds les plus en bas de l'arbre primaire (les nœuds feuilles d'abord) et en remontant jusqu'au nœud source.

Les simulations montrent que cette technique de protection remontante présente un taux de protection légèrement inférieur au taux idéal et très supérieur à celui correspondant à la technique de protection par forêt duale.

**Mots Clés :** réseau, multicast, arbre multicast, protection par forêt duale, protection remontante.

## 1 INTRODUCTION

Le multicast est la technique qui permet d'envoyer des données à différents destinataires sans que celles-ci ne soient dupliquées sur les liens du réseau. Une portion importante de la bande passante réseau est sauvée par l'utilisation d'une structure d'arbre, dit arbre de distribution ou arbre multicast.

Avec l'augmentation de l'intérêt pour les communications de groupes actuellement et la multiplication des applications très consommatrices de la bande passante telles que la TV numérique et le jeu en réseau, le multicast est devenu très utile. Ainsi et au lieu d'établir une connexion unicast pour chaque couple (source, destination), une seule connexion multicast évitant la duplication des mêmes données (multicast) sur

les liens du réseau, grâce à l'utilisation d'un arbre pour le routage, est utilisée.

Vu que la plupart des applications multicast sont de type temps réel et sont sensibles aux ruptures de communications, les réseaux actuels doivent réagir très vite aux pannes afin de maintenir les communications. Pour ce faire, différentes techniques de protection ont été développées. Ces dernières ont pour rôle de déterminer des chemins de secours permettant de remplacer les chemins primaires des communications affectées par une panne.

Les techniques de protection peuvent être classées en deux catégories : réactive et proactive. Dans les techniques de protection réactive, les chemins de secours ne sont calculés qu'après la survenue d'une panne alors que dans les techniques de protection proactive, les chemins de secours sont pré-calculés et généralement préconfigurés à l'avance. Les techniques de la première catégorie ont l'avantage de résister aux changements de la topologie et de ne pas induire de surcoût lié à la maintenance des communications. Cependant, elles présentent le désavantage de délais de récupération élevés et indésirables pour de nombreuses applications. Afin de diminuer sensiblement les délais de récupération, la tendance actuelle est d'utiliser les techniques de protection proactive qui permettent en plus, de pré-réserver les ressources afin d'assurer leur disponibilité (même qualité de service) sur les chemins de secours après la récupération d'une panne.

Bien qu'il existe aujourd'hui une multitude de techniques de protection proactive unicast [Meyer, 2004] [Murakami, 1998] [Pan, 2005] [Ramamurthy, 1999], ces dernières ne sont pas facilement adaptables au multicast. En effet, l'utilisation d'une technique de protection proactive unicast pour protéger des communications multicast ne permet pas de se prémunir contre le risque d'apparition de boucles et/ou de duplication de trafic sur quelques liens du réseau (après la récupération). Un exemple d'adaptation d'une technique de protection unicast pour la protection multicast et résultant en la formation d'une boucle est donné en [Saidi, 2006]. De plus, les techniques inspirées de l'unicast induisent des coûts élevés des structures de secours utilisées pour la protection.

Afin de pallier les désavantages des techniques précédentes, d'autres travaux ont porté sur la recherche

d'une structure de routage moins coûteuse et plus appropriée au multicast. Ainsi et durant ces dernières années, quelques techniques de protection proactive destinées exclusivement à protéger les communications multicast ont vu le jour [Fei, 2000] [Medard, 1999] [Saidi, 2006]. Ces dernières tendent à construire une structure de routage (en général un arbre ou une forêt) interconnectant un sous-ensemble des nœuds de l'arbre primaire sans utilisation des liens et des autres nœuds de cet arbre primaire. Une telle structure de routage a pour rôle de fournir les chemins de secours destinés à remplacer les chemins primaires de la communication multicast affectés par une panne.

Si ces dernières techniques simplifient considérablement le rétablissement des communications, elles ne permettent pas cependant d'optimiser la protection. En effet, certains composants de l'arbre primaire ne peuvent pas être protégés avec ces techniques de protection bien que la topologie soit assez redondante pour le permettre. Cela est souvent le cas dans les topologies faiblement maillées où la suppression de tous les liens et de certains nœuds de l'arbre primaire lors de la recherche des routes de secours (pas d'exploration de tout l'arbre primaire) résulte en l'élimination d'un certain nombre de routes de secours potentielles.

Afin de mieux protéger les communications multicast, nous proposons ici une nouvelle technique de protection proactive multicast dite protection remontante. Cette technique tente d'interconnecter toutes les branches de l'arbre primaire en commençant par les nœuds feuilles et en remontant jusqu'à la source. C'est une technique qui utilise des bouts de chemins de l'arbre primaire afin de construire les boucles de protection.

Dans ce qui suit, nous allons donner un aperçu en section 2 des techniques de protection proactive destinées exclusivement à protéger les communications multicast, puis nous proposons et décrivons en section 3 la technique de protection remontante. En section 4, nous présenterons le modèle de simulation utilisé pour évaluer les performances de la technique de protection remontante, puis nous analyserons les résultats obtenus. La dernière section sera consacrée aux conclusions.

## 2 ETAT DE L'ART

Deux approches de protection proactive peuvent être adoptées pour protéger les communications multicast :

1. Découpage de l'arbre primaire en plusieurs chemins (ou segments) et utilisation d'une technique de protection proactive unicast ;
2. Utilisation d'une technique de protection proactive spécifique au multicast.

Concernant le premier type d'approche, l'arbre primaire peut être découpé en chemins ou en segments, chacun sera protégé indépendamment en utilisant une technique de protection proactive unicast. Par exemple, les chemins de la source vers les différents membres multicast peuvent être protégés en utilisant la technique de protection par chemins disjoints [Meyer, 2004] [Saidi, 2006].

Ce type d'approche de protection n'induit aucun surcoût lié au développement mais il présente deux inconvénients restreignant son utilisation :

1. Risque de formation de boucles après la récupération [Saidi, 2006] ;
2. Coût élevé des structures de secours.

Afin de pallier les deux désavantages ci-dessus, la deuxième approche a été explorée et d'autres techniques de protection proactive multicast ont été mises en œuvre. Nous présentons ci-dessous deux principales techniques de cette approche qui sont la protection par arbre redondant et la protection par forêt duale.

Dans tout ce qui suit, la topologie du réseau sera représentée par un graphe  $G$ . L'ensemble des destinations  $\{Mi\}_{i>0}$  et la source multicast  $S$  forment l'ensemble des membres multicast  $M$ . L'arbre primaire (orienté) est noté  $T_p$ .

### 2.1 Arbre redondant de protection

Dans cette technique de protection, un arbre dit redondant et couvrant tous les nœuds membres multicast ( $M$ ) est pré-calculé (et souvent pré-configuré) pour protéger l'arbre primaire. Mis à part les nœuds membres, l'arbre redondant ne doit contenir aucun autre nœud ou lien de l'arbre primaire [Medard, 1999].

Sur la figure 1, l'arbre primaire composé des arcs  $(S, M1)$ ,  $(M1, M2)$  et  $(M1, M3)$  est protégé par un arbre redondant disjoint et couvrant tous les nœuds membres multicast  $\{S, M1, M2, M3\}$ . Cet arbre redondant est composé des arcs  $(S, A)$ ,  $(A, M1)$ ,  $(A, B)$ ,  $(B, M2)$  et  $(B, M3)$ .

Lorsqu'une panne est détectée, la source qui doit être informée redirigera le trafic de l'arbre primaire vers l'arbre redondant.

Cette technique permet de traiter un nombre quelconque de pannes simultanées mais elle présente divers inconvénients :

1. Elle nécessite un ratio de redondance du graphe élevé pour être applicable. Cela induit un taux de protection très faible.
2. L'information sur la panne doit remonter jusqu'à la source.
3. Le coût de l'arbre redondant est élevé.

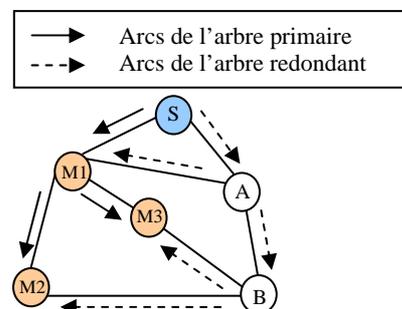


Fig. 1. Arbre redondant de protection

### 2.2 Forêt duale de protection

Afin d'augmenter la probabilité d'existence de la structure de secours et pour diminuer de son coût, [Fei, 2000] propose de construire un arbre de secours, dit arbre dual, couvrant seulement l'ensemble des nœuds feuilles de l'arbre primaire. Cette technique de

protection permet le rétablissement des communications multicast lors des pannes des liens si l'arbre dual couvre tous les nœuds feuilles mais elle n'est pas sûre dans le cas de pannes des nœuds.

Pour pouvoir traiter convenablement les pannes des nœuds, [Saidi, 2006] propose la technique de protection par forêt duale qui permet aussi d'augmenter sensiblement le taux de protection. Dans cette technique, les chemins de secours forment une forêt qui couvre l'ensemble des nœuds feuilles de l'arbre primaire. Cette forêt est bidirectionnelle (nécessite des liens bidirectionnels) et doit être composée d'un nombre minimum d'arbres.

Sur la figure 2 (a), l'arbre primaire composé des arcs (S, A), (S, B), (A, M1), (A, C), (C, M2), (B, D), (D, M3) et (B, M4) est protégé par la forêt qui consiste en les deux arêtes (M1, M2) et (M3, M4). Lorsqu'une panne est détectée, seuls les chemins appartenant à la forêt duale peuvent être utilisés pour rétablir la communication multicast.

Chaque nœud de l'arbre primaire prédétermine le chemin de secours appartenant à la forêt duale qui permet la récupération dans le cas de la panne de son nœud père dans l'arbre primaire ou du lien menant à ce nœud. Pour ce faire, il divise l'ensemble des nœuds feuilles de l'arbre primaire en trois sous-ensembles : ensemble des nœuds sûrement affectés (SA), ensemble des nœuds possiblement affectés (PA) et ensemble des nœuds non affectés (NA).

L'ensemble  $SA_x$  d'un nœud  $x$  consiste en l'ensemble des nœuds feuilles du sous-arbre primaire dont il est racine. Son ensemble  $PA_x$  est composé des nœuds feuilles du sous-arbre primaire dont la racine est le père du nœud  $x$  dans l'arbre primaire et qui n'appartiennent pas à  $SA_x$ . L'ensemble  $NA_x$  consiste en le reste des nœuds feuilles de l'arbre primaire. Pour le nœud  $C$  de la figure 2 (a) par exemple, ses ensembles consistent en :  $SA_C = \{M2\}$ ,  $PA_C = \{M1\}$  et  $NA_C = \{M3, M4\}$ .

Le chemin de secours prédéterminé par un nœud  $x$  sera celui qui interconnecte dans la forêt duale un nœud de  $SA_x$  à un nœud de  $NA_x$  ou à défaut à un nœud de  $PA_x$ . Pour le nœud  $C$  de la figure 2 (a), le chemin (M2, M1) sera sélectionné vu qu'aucun chemin n'interconnecte dans la forêt duale un nœud de  $SA_C$  à un nœud de  $NA_C$ .

Lorsqu'une panne est détectée par un nœud, ce dernier active son chemin de secours en envoyant le message

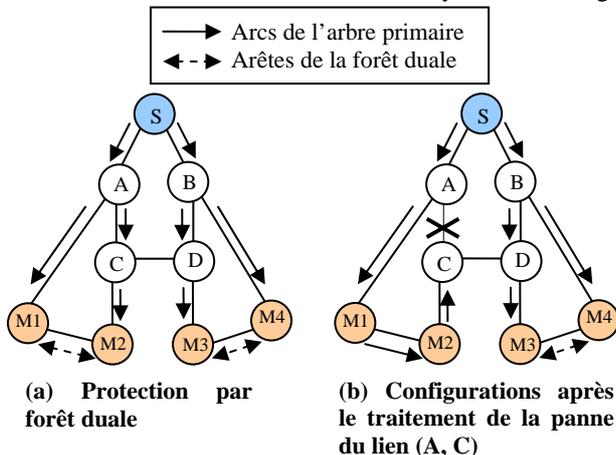


Fig. 2. Forêt duale de protection

*RECONFIG*. Typiquement, si le nœud  $C$  détecte une panne sur son interface père, il envoie le message *RECONFIG* au nœud  $M1$  (extrémité non affectée du chemin de secours du nœud  $C$ ). Ce message passe par  $M1$  et provoque l'inversion des arcs sur la partie primaire et l'installation des entrées de routage multicast sur la partie du chemin de secours (figure 2 (b)).

Cette technique de protection permet de résoudre le problème de formation des boucles par l'utilisation d'une forêt duale dont le coût n'est pas élevé. Cependant et particulièrement dans le cas de topologies faiblement maillées, le taux de protection de cette technique se dégrade et quelques parties de l'arbre primaire risquent de ne pas être protégées. Typiquement et sur la figure 2 (a), aucun des liens (S, A) et (S, B) ou des nœuds  $A$  et  $B$  n'est protégé bien que la topologie le permette.

### 3 PROTECTION REMONTANTE

Dans une technique de protection efficace, la probabilité qu'une communication multicast, affectée par une panne, soit complètement rétablie est élevée et très proche de l'idéal. En d'autres termes, une technique de protection efficace doit être en mesure de fournir des chemins de secours permettant le rétablissement complet de la communication multicast s'il est possible de déterminer un nouvel arbre couvrant tous les membres après la panne.

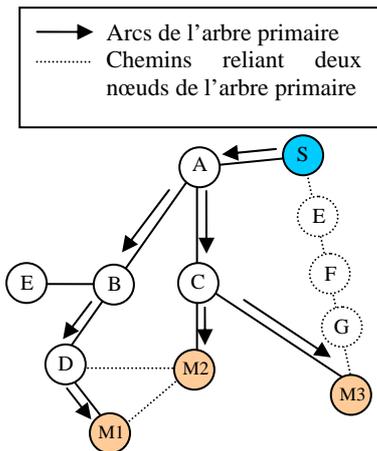
Si la technique de protection par forêt duale présente divers avantages, son taux de protection est faible pour certains types de topologies (plus particulièrement les topologies faiblement maillées) et peut être amélioré en explorant tous les nœuds de l'arbre primaire et de la topologie. Sur la figure 2 (a) par exemple, l'ajout du chemin (C, D) à la forêt duale permet la protection de tous les nœuds (non membre multicast) et de tous les liens de l'arbre primaire.

Dans le reste de cette section, nous allons montrer comment choisir une structure de routage de secours donnant un taux de protection élevé. Nous commençons en section 3.1 par désigner la partie de l'arbre primaire qui est protégée par l'ajout d'un chemin interconnectant deux nœuds de l'arbre primaire, puis nous présenterons en section 3.2 la technique de protection remontante qui est basée sur les conclusions de la section 3.1. Dans la dernière sous-section, nous adapterons l'algorithme de restauration de la forêt duale à la protection remontante pour permettre le rétablissement des communications multicast après la survenue d'une panne.

#### 3.1 Partie protégée par une route de secours reliant deux nœuds de l'arbre primaire

L'ajout d'une route reliant deux nœuds de l'arbre primaire résulte en la création d'une boucle. Cette dernière permet la protection de la communication multicast contre les pannes de certains nœuds de la boucle et de tous ses liens.

Sur la figure 3, l'ajout de la route (D, M2) à l'arbre primaire provoque la création de la boucle (A, B, D, M2, C, A). La communication multicast (ayant comme



**Fig. 3. Protection fournie par une boucle**

source  $S$  et comme destinations  $M1$ ,  $M2$  et  $M3$ ) est protégée contre la panne de n'importe quel lien de l'arbre primaire appartenant à la boucle puisqu'il est toujours possible d'atteindre tous les nœuds de la boucle à partir du nœud  $A$  (qui est l'ancêtre de tous les nœuds de la boucle) via deux chemins complètement disjoints. Si le lien  $(A, C)$  tombe par exemple, il sera possible d'atteindre le nœud  $C$  (et tous les autres nœuds de la boucle) par le chemin  $(A, B, D, M2, C)$ . Lors d'une panne du lien  $(A, C)$ , le remplacement des deux branches de l'arbre primaire  $(A, B, D)$  et  $(A, C, M2)$  par la branche  $(A, B, D, M2, C)$  permet la construction d'un nouvel arbre multicast couvrant tous les nœuds membres et rétablissant la communication multicast.

Concernant la protection des nœuds, nous signalons que la communication multicast n'est protégée contre la panne d'aucun des deux nœuds extrémités de la route reliant les deux nœuds de l'arbre primaire, ni contre la panne du nœud ancêtre de tous les autres nœuds de la boucle formée. Sur la figure 3, la boucle  $(A, B, D, M2, C, A)$  ne protège pas contre la panne des nœuds  $D$ ,  $M2$  (extrémités de la route) et  $A$  (ancêtre de tous les nœuds de la boucle). Par contre, Elle permet de protéger entièrement ou partiellement les autres nœuds de la boucle. Ainsi, la boucle  $(A, B, D, M2, C, A)$  permet de protection entière contre la panne du nœud  $B$  (car l'unique fils de  $B$  dans l'arbre primaire appartient à la boucle) et une protection partielle contre la panne du nœud  $C$  (car le nœud  $M3$  qui est un des deux fils de  $C$  dans l'arbre primaire n'appartient pas à la boucle). Pour offrir une protection entière contre la panne du nœud  $C$ , il faut déterminer une route de secours (boucle) reliant le nœud  $M3$  ou un de ses descendants à un nœud qui ne sera pas affecté par une panne qui toucherait le nœud  $C$ .

Sur la figure 3, l'ajout de la route de secours  $(S, E, F, G, M3)$  permet la protection entière du nœud  $C$  vu que cette route interconnecte le nœud  $M3$  à un nœud  $S$  qui ne peut pas être affecté par une panne touchant le nœud  $C$ .

Si l'on rajoute maintenant le chemin  $(M1, M2)$  à l'arbre primaire, une troisième boucle  $(A, B, D, M1, M2, C)$  sera formée. Cette dernière contient tous les nœuds appartenant à l'arbre primaire et à la première boucle  $(A, B, D, M2, C)$ . En conséquence, la protection fournie par la troisième boucle englobe celle fournie par la première boucle. Concrètement, la troisième boucle permet la

protection contre la panne du nœud  $D$  et du lien  $(D, M1)$  en plus des composants protégés par la première boucle.

Pour des raisons de coût de la structure de secours, il serait plus intéressant de ne configurer que les boucles non englobées. Sur la figure 3, la route  $(D, M2)$  ne devrait pas être configurée en secours car il est plus préférable de configurer l'autre route  $(M1, M2)$  protégeant un ensemble de composants contenant celui protégé par la route  $(D, M2)$ .

### 3.2 Structure de secours

La technique de protection remontante tend à déterminer les boucles englobantes afin de construire les chemins de secours. Pour cela, elle explore l'arbre multicast du bas vers le haut (des nœuds feuilles jusqu'au nœud source) pour trouver des boucles résultant de l'interconnexion de deux nœuds de l'arbre primaire. Toute boucle formée sera fusionnée avec une autre si elles partagent au moins un lien.

Après la construction de l'arbre primaire, l'algorithme de construction de la structure de secours (avec la technique de protection remontante) détermine l'ensemble des nœuds feuilles de cet arbre. Sur la figure 4 (a), un arbre primaire couvrant le groupe multicast  $\{S, M1, M2, M3, M4, M5\}$  dans le graphe  $G$  est construit.

Pour obtenir des boucles non englobées par d'autres, il est intéressant de n'explorer que les nœuds feuilles de l'arbre primaire dans un premier temps. Pour cela, un nouveau graphe de secours  $G_s$  est déduit du graphe d'origine  $G$  par l'élimination de tous les nœuds internes et liens de l'arbre primaire (figure 4 (a) et figure 4 (b)). Cette transformation garantit que toute route reliant deux nœuds de l'arbre primaire dans  $G_s$  est forcément une route interconnectant deux nœuds feuilles de l'arbre primaire dans  $G$  et par conséquent, elle produit une boucle non englobée. Ensuite, tous les nœuds de l'arbre primaire (les feuilles) seront interconnectés dans  $G_s$  par des routes de secours.

A chaque nœud  $x$  du graphe  $G$  est attribué un identifiant de boucle  $numLoop[x]$  indiquant la boucle (ou la composante connexe) à laquelle il appartient. Ainsi et initialement, tous les nœuds feuilles de l'arbre primaire appartenant à la même composante connexe dans  $G_s$  auront un même identifiant de boucle différent de  $\infty$ . Les autres nœuds du graphe  $G$  auront  $\infty$  comme identifiant de boucle. En conséquence, les nœuds du graphe  $G$  de la figure 4 (b) auront initialement les identifiants de boucle suivants :  $numLoop[M1] = numLoop[M2] = 0$  ;  $numLoop[M3] = 1$  ;  $numLoop[M4] = 2$  ;  $numLoop[M5] = 3$  ;  $numLoop[A] = numLoop[B] = numLoop[C] = numLoop[D] = numLoop[S] = \infty$ . Les valeurs de  $numLoop$  associées aux nœuds de  $G_s$  à chaque étape de la construction des routes de secours sont représentées par des nombres (placés à côté des nœuds) sur la figure 4.

Après cette étape, l'arbre primaire est parcouru du bas vers le haut (à partir des nœuds feuilles jusqu'à la source) et les couples  $\langle nœud\_père, nœud\_fils \rangle$  de l'arbre primaire seront insérés (suivant l'ordre de parcours) ou supprimés d'une liste  $lien\_arbre\_primaire$  suivant les deux règles suivantes :

---

**Alg 1. Construction des routes de secours**

---

**Entrées :**

**Graphe :**  $G = (V, E)$  {graphe correspondant à la topologie}

**Arbre :**  $T_p$  {arbre primaire couvrant l'ensemble des nœuds membres}

**Sorties :**

**Ensemble Lien :**  $S$  {liens appartenant aux routes de secours qui forment la forêt de secours  $F_p$ }

**Variabiles :**

**Tableau Identifiant [V] :** numLoop {à chaque nœud du graphe est associé un numéro de boucle à laquelle il appartient}

**Nœud :**  $x, y$

**Liste <Nœud, Nœud> :** liens\_arbre\_primaire

**Booléen :** arrêt

**Ensemble Nœud :** nœuds\_feuilles

**Graphe :**  $G_s = (V_s, E_s)$

**Entier :** nombre\_composantes {nombre de composantes à interconnecter dans le graphe pour obtenir des routes de secours}

**Début**

- $G_s = \text{graphe\_des\_secours}(G, T_p)$   
{ $G_s$  est obtenu en éliminant de  $G$  tous les nœuds internes et tous les liens de l'arbre primaire  $T_p$ }
- nœuds\_feuilles = feuilles( $T_p$ )  
{L'ensemble des nœuds feuilles de l'arbre primaire est affecté à l'ensemble *nœuds\_feuilles*}
- Pour** tout ( $x \in G$ ) numLoop =  $\infty$  **FinPour**  
**Pour** tout ( $x \in \text{nœuds\_feuilles}$ )  
numLoop [ $x$ ] = **nouvel\_identifiant**()  
{**nouvel\_identifiant**() retourne un identifiant fini et différent de tous les précédents à chaque nouvel appel}  
**FinPour**
- arrêt = faux  
nombre\_composantes = | nœuds\_feuilles |
- Tant que** (nombre\_composantes > 1 **et** non arrêt)  
 $\langle x, y \rangle =$   
**interconnecter\_deux\_composantes**  
(numLoop,  $G_s$ )  
{détermine un ensemble de liens permettant d'interconnecter deux nœuds  $x$  et  $y$  dans  $G_s$  et vérifiant (numLoop[ $x$ ] != numLoop[ $y$ ] **et** numLoop[ $x$ ] <  $\infty$  **et** numLoop[ $y$ ] <  $\infty$ )}  
**Si**  $\langle x, y \rangle = \langle \text{NULL}, \text{NULL} \rangle$  **alors**  
{plus de route interconnectant deux nœuds feuilles}  
arrêt = vrai  
**Sinon**  
 $S = S \cup \text{route}(x, y, G_s)$  {la fonction **route** (Nœud  $x$ , Nœud  $y$ , Graph  $G_s$ ) retourne les liens de la route permettant d'interconnecter dans  $G_s$  le nœud  $x$  au nœud  $y$ }

---

**Alg 1. Suite**

---

nombre\_composantes =

nombre\_composantes - 1

**mettre\_à\_jour\_numLoop**

(numLoop[ $x$ ], numLoop[ $y$ ],  $G_s$ )

{tout nœud  $a$  du graphe  $G_s$  dont  
numLoop[ $a$ ] = numLoop[ $x$ ] exécutera  
numLoop[ $a$ ] = numLoop[ $y$ ]}

**Finsi**

**FinTantque**

- Si** nombre\_composantes < 2 **alors**

**aller à** étape 9

**Finsi**

- liens\_arbre\_primaire = **ordonner\_liens** ( $T_p$ )  
{l'arbre primaire  $T_p$  est parcouru du bas vers le haut, un seul passage par nœud est autorisé. Tout couple de nœuds  $\langle x, y \rangle$  parcouru successivement ( $x$  est le père de  $y$  dans l'arbre) est inséré en queue de la liste retournée par la fonction **ordonner\_liens** (Arbre  $T$ )}
- Tant que** (nombre\_composantes > 1)

$\langle x, y \rangle =$  **extraire\_tête**

(Liens\_arbre\_primaire)

{extrait le premier couple de nœuds de la liste *liens\_arbre\_primaire* et le retourne}

**Si**  $\langle x, y \rangle = \langle \text{NULL}, \text{NULL} \rangle$  **alors**

{liste vide}

**aller à** étape 9

**Finsi**

numLoop[ $x$ ] = numLoop[ $y$ ]

**rajouter** ( $x, G_s$ ) {le nœud  $x$  est rajouté à  $G_s$  ainsi que tous ses liens adjacents n'appartenant pas à l'arbre primaire}

$z =$  **interconnecter\_à\_composante**

( $x, \text{numLoop}$ )

{retourne un nœud  $z$  appartenant à l'arbre primaire pouvant être relié au nœud  $x$  dans  $G_s$  et vérifiant (numLoop[ $z$ ] <  $\infty$  **et** numLoop[ $z$ ] != numLoop[ $x$ ])}

**Tant que** ( $z \neq \text{NULL}$ )

{il existe un nœud  $z$  relié à  $x$  dans  $G_s$ }

$S = S \cup \text{route}(x, z, G_s)$

nombre\_composantes =

nombre\_composantes - 1

**mettre\_à\_jour\_numLoop**

(numLoop[ $z$ ], numLoop[ $x$ ],  $G_s$ )

$z =$  **interconnecter\_à\_composante**

( $x, \text{numLoop}$ )

**FinTantque**

**FinTantque**

- construire\_forêt** ( $S, T_p$ )

{**construire\_forêt** ( $S, T_p$ ) permet de construire une forêt à partir des liens de  $S$  en éliminant toutes les boucles. Les nœuds feuilles de chaque arbre appartenant à la forêt doit appartenir à l'arbre primaire  $T_p$ }

---

1. Ordre : un couple de nœuds  $\langle x, y \rangle$  n'est inséré dans la liste que si tous les couples  $\langle y, z_1 \rangle, \langle y, z_2 \rangle, \dots, \langle y, z_n \rangle$  de l'arbre primaire ont été déjà rajoutés à la liste. Pour l'exemple de la figure 4, la liste *lien\_arbre\_primaire* consistera en :  $[\langle S, M1 \rangle, \langle B, M2 \rangle, \langle B, M3 \rangle, \langle C, M4 \rangle, \langle A, M5 \rangle, \langle B, C \rangle, \langle S, A \rangle, \langle S, B \rangle]$ .
2. De cette liste *lien\_arbre\_primaire*, tous les couples  $\langle x_2, y \rangle, \dots, \langle x_n, y \rangle$  suivant le couple  $\langle x_1, y \rangle$  dans la liste seront supprimés. En conséquence, les couples  $\langle M3, B \rangle$  et  $\langle C, B \rangle$  seront supprimés de la liste (car précédés du couple  $\langle M2, B \rangle$ ). La liste finale *lien\_arbre\_primaire* sera alors :  $[\langle S, M1 \rangle, \langle B, M2 \rangle, \langle C, M4 \rangle, \langle A, M5 \rangle, \langle S, A \rangle, \langle S, B \rangle]$ .

Ensuite, l'étape de la recherche des chemins de secours sera entamée. Pour chaque couple  $\langle x, y \rangle$  de la liste finale,  $x$  prendra le numéro de boucle du nœud  $y$

( $numLoop[x] = numLoop[y]$ ) et sera rajouté ainsi que tous ses liens adjacents et n'appartenant pas à l'arbre primaire au graphe  $G_s$ . Le nœud  $x$  est exploité par la suite afin de déterminer tous les chemins (de secours) dans  $G_s$ , l'interconnectant à des nœuds de l'arbre primaire dont le numéro de boucle est différent du sien. Sur la figure 4 (c) est illustrée  $G_s$  après le rajout des nœuds  $S, B$  et  $C$ . On remarque que le nœud  $C$  peut être relié à deux nœuds :  $M3$  et  $M4$ . Vu que  $numLoop[M4] = numLoop[C]$ , la route reliant les deux nœuds  $M4$  et  $C$  ne sera pas ajoutée à l'ensemble des routes de secours. Par contre et comme  $numLoop[M3] \neq numLoop[C]$ , la route  $(C, M3)$  sera rajoutée à l'ensemble des routes de secours et tous les nœuds de l'arbre primaire ayant le même identifiant de boucle que  $M3$  changeront leur identifiant de boucle en  $numLoop[C]$  (figure 4 (d)).

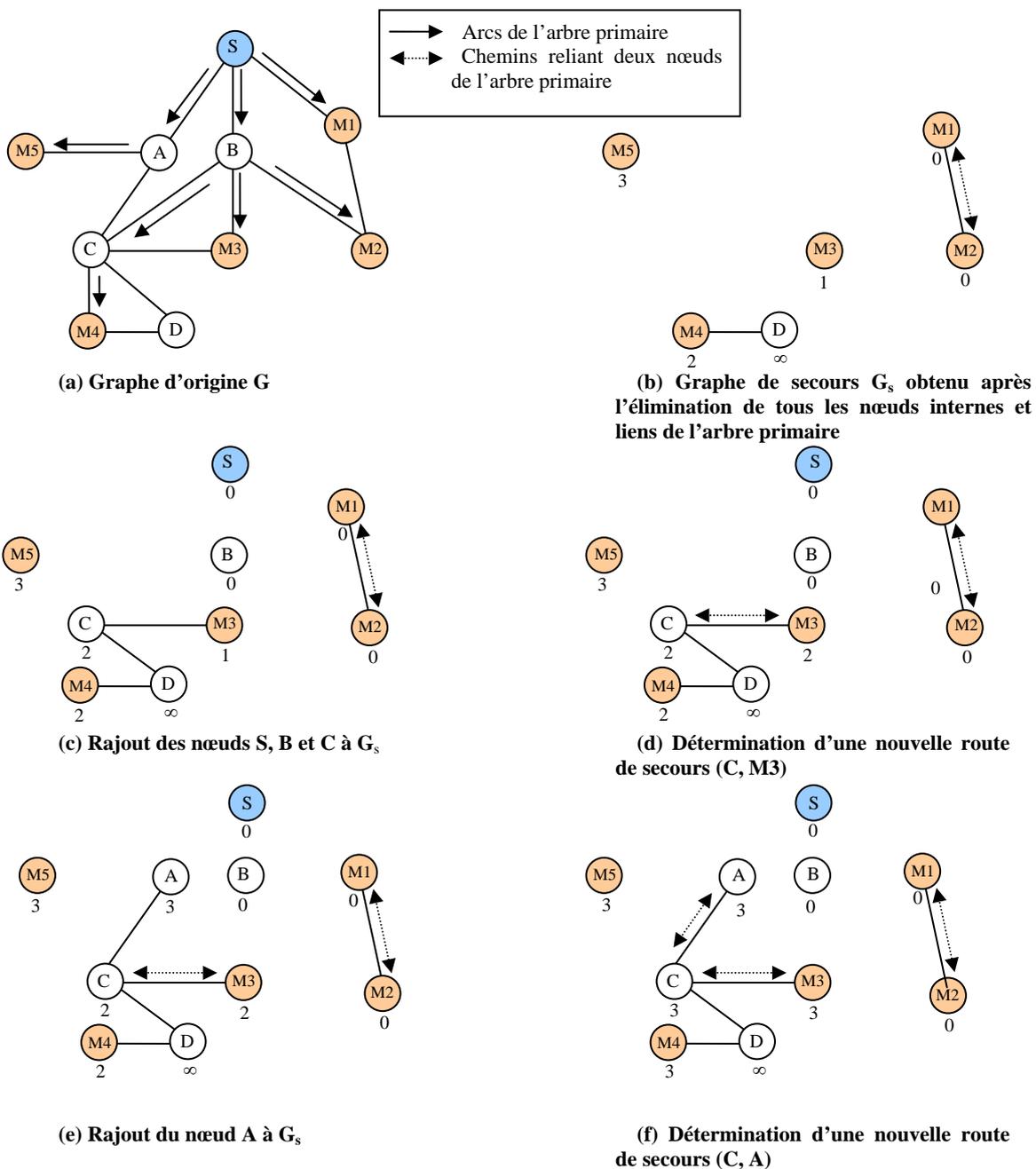


Fig. 4. Protection remontante

De même, le nœud  $A$  sera rajouté à  $G_s$  (figure 4 (e)) et la route  $(C, A)$  sera ajoutée à l'ensemble des routes de secours (figure 4 (f)).

Formellement, l'algorithme de construction des routes de secours est résumé en les étapes de l'algorithme 1.

---

**Alg 2. Algorithme de restauration exécuté par un nœud  $x$  détectant une panne de son interface père**

---

1. **déduire\_ensembles** ( $x, T_p, SA_x, SP_x, PA_x, NA_x$ )  
{Le nœud  $x$  divise l'ensemble des nœuds feuilles de l'arbre primaire  $T_p$  en quatre sous-ensembles  $SA_x, SP_x, PA_x, NA_x$ }
  2. **contracter\_frères** ( $x, T_p, G, G_c, PA_x, PA_x^c$ )  
{Contracter les sous-arbres ayant comme racines des nœuds frères de  $x$  dans l'arbre primaire  $T_p$ . Un nouveau graphe  $G_c$  est alors obtenu suite à la contraction de ces nœuds de  $G$ . L'ensemble  $PA_x^c$  englobe les nœuds contractés dans  $G_c$  auquel correspond l'ensemble des nœuds de  $PA_x$  dans le graphe d'origine  $G$ }
  3. **déduire\_chemin\_secours**  
( $F_p^c, SA_x, SP_x, PA_x^c, NA_x, bp_x^c, G_c$ )  
{Déduire un plus court chemin de secours  $bp_x^c$  appartenant à la forêt de protection  $F_p^c$  contractée (dans  $G_c$ ) qui interconnecte un nœud de  $SA_x$  à un nœud de  $NA_x$ .  
Si un tel chemin n'existe pas, un plus court chemin de secours de  $F_p^c$  interconnectant un nœud de  $SA_x$  à un nœud (contracté) de  $PA_x^c$  est assigné à  $bp_x^c$ .  
Sinon un chemin de secours de  $F_p^c$  interconnectant un nœud de  $SA_x$  au nœud père de  $x$  est affecté à  $bp_x^c$ .  
A défaut,  $bp_x^c$  sera égal à *chemin infini*}
  4. **Si**  $bp_x^c = \text{chemin infini}$  **alors**  
{Le rétablissement de la connexion avec la technique de protection remontante est impossible}  
**aller à fin**
- Finsi**
5. **diviser\_chemin\_secours** ( $PA_x^c, bp_x^c$ )  
{Si  $bp_x^c$  inclue un nœud de  $PA_x^c$  alors le plus petit sous-chemin interconnectant l'extrémité de  $bp_x^c$  appartenant à  $SA_x$  au plus proche nœud de  $PA_x^c$  est affecté à  $bp_x^c$ .  
Ne rien faire sinon}
  6. **décontracte\_chemin** ( $bp_x^c, G_c, G, bp_x$ )  
{Le chemin de secours  $bp_x$  qui est la projection de  $bp_x^c$  dans  $G$  est déduit}
  7. **extremities** ( $bp_x, e_1, e_2$ )  
{L'extrémité du chemin de secours  $bp_x$  appartenant à  $SA_x$  est retournée dans le paramètre  $e_1$  la seconde extrémité du chemin de secours  $bp_x$  est retournée dans le paramètre  $e_2$ }
  8. **créer\_msg\_Reconfig** ( $x, T_p, bp_x, RECONFIG$ )  
{Tous les nœuds sur le chemin appartenant à  $T_p$  et reliant  $x$  à l'extrémité  $e_1$  seront listés dans le message  $RECONFIG$  dans cet ordre.  
Ces nœuds seront suivis des nœuds de  $bp_x$  (de  $e_1$  à  $e_2$ ) dans cet ordre aussi}
  9. **envoyer\_msg\_Reconfig** ( $x, RECONFIG$ )  
{Le message  $RECONFIG$  est envoyé au nœud  $succ(x, RECONFIG)$  qui est le nœud successeur de  $x$  dans la liste incluse dans le message  $RECONFIG$ }
- 

---

**Alg 2. Suite**

---

10. **père**( $x, T_p$ )  $\leftarrow succ(x, RECONFIG)$   
{Le successeur de  $x$  dans la liste incluse dans le message  $RECONFIG$  devient son nouveau père dans l'arbre multicast}
  11. **supprimer\_fils** ( $x, T_p, succ(x, RECONFIG)$ )  
{Le nœud  $succ(x, RECONFIG)$  est supprimé de la liste des enfants de  $x$  dans  $T_p$ }
- 

---

**Alg 3. Algorithme de restauration exécuté par un nœud  $y$  recevant le message  $RECONFIG$**

---

1. **Si**  $y \neq e_2$  **alors**  
{ $e_2$  est le dernier nœud de la liste incluse dans le message  $RECONFIG$ }  
**envoyer\_msg\_RECONFIG**( $y, RECONFIG$ )  
{Le message  $RECONFIG$  est envoyé au nœud  $succ(y, RECONFIG)$ }  
**père**( $y, T_p$ )  $\leftarrow succ(y, RECONFIG)$   
{Le père de  $y$  dans  $T_p$  devient le nœud  $succ(y, RECONFIG)$ }  
**supprimer\_fils** ( $y, T_p, succ(y, RECONFIG)$ )  
{Le nœud  $succ(y, RECONFIG)$  est supprimé de la liste des enfants de  $y$  dans  $T_p$ }
- endif**
2. **ajouter\_fils**( $y, T_p, pred(y, RECONFIG)$ )  
{Le nœud  $pred(y, RECONFIG)$  qui est le prédécesseur de  $y$  dans la liste incluse dans le message  $RECONFIG$  est ajouté à la liste des enfants de  $y$  dans  $T_p$ }
- 

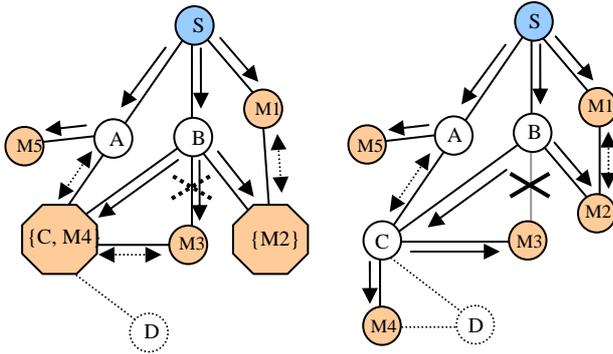
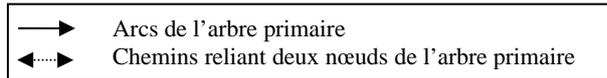
**3.3 Algorithme de restauration**

Après la détermination de l'ensemble des routes de secours, chaque nœud doit déterminer la route qu'il utiliserait en cas de panne. Pour ce faire, les nœuds exécutent un algorithme de restauration permettant de déterminer la route la plus sûre de toutes celles qui appartiennent à la structure de secours. Pour ce faire, chaque nœud  $x$  (différent de la source) de l'arbre primaire divise l'ensemble des nœuds de l'arbre primaire en 4 sous-ensembles : ensemble des nœuds affectés  $AF_x$ , singleton père  $SP_x$ , ensemble des nœuds possiblement affectés  $PA_x$  et ensemble des nœuds non affectés  $NA_x$ .

L'ensemble  $SA_x$  est composé de tous les nœuds du sous-arbre ayant  $x$  comme racine. Le singleton père  $SP_x$  contient le nœud père du nœud  $x$ . L'ensemble  $PA_x$  consiste en tous les nœuds du sous-arbre dont la racine est le père du nœud  $x$  (dans l'arbre primaire) et qui n'appartiennent pas à l'ensemble  $(SA_x \cup SP_x)$ . L'ensemble  $NA_x$  consiste en le reste des nœuds de l'arbre primaire.

Pour l'exemple de la figure 4, les quatre sous-ensembles déterminés par le nœud  $M3$  consisteront en :  $SA_{M3} = \{M3\}$ ,  $SP_{M3} = \{B\}$ ,  $PA_{M3} = \{M2, C, M4\}$  et  $NA_{M3} = \{S, A, M1, M5\}$ .

Pour simplifier la compréhension de l'algorithme de la restauration, nous contractons tous les nœuds de chaque sous-arbre ayant comme racine un nœud frère du nœud  $x$  en un seul nœud. Pour l'exemple de la figure 4, les nœuds  $C$  et  $M4$  seront contractés en le nœud  $\{C, M4\}$  et le nœud  $M2$  sera contracté en le nœud  $\{M2\}$  (figure 5



(a) Graphe  $G_c$  déduit par contraction des sous-arbres de racines C et M2

(b) Situation après la récupération de la panne du lien (B, M3)

**Fig. 5. La restauration avec la protection remontante**

(a). Après cette étape de contraction, le nœud  $x$  cherchera à déterminer un plus court chemin appartenant à la forêt de secours (dans le graphe contracté) permettant d’interconnecter un nœud  $y$  de  $SA_x$  à un nœud de  $NA_x$  ou à défaut à un nœud de  $PA_x$  sinon au nœud de  $SP_x$ . Si le chemin déterminé (reliant  $x$  à un nœud  $NA_x$ ) passe par un nœud de  $PA_x$  alors seule la partie du chemin allant du nœud de  $SA_x$  jusqu’au plus proche nœud de  $PA_x$  est retournée. Si aucun chemin n’existe alors la récupération n’est pas rendue avec la protection remontante.

Dans l’exemple de la figure 5 (b), le nœud  $M3$  déterminera l’unique chemin ( $M3, \{C, M4\}, A$ ) permettant de relier le nœud  $M3$  de  $SA_{M3}$  au nœud  $A$  de  $NA_{M3}$ . A ce chemin dans le graphe contracté correspond le chemin ( $M3, C, A$ ) dans le graphe d’origine. Ce chemin passe par un nœud de  $PA_{M3}$  alors seule la partie du chemin ( $M3, C$ ) est retournée.

Enfin et lorsque la panne est détectée, le nœud  $x$  enverra un message *RECONFIG* au nœud extrémité de la route de secours  $z$  (l’autre nœud extrémité de la route de secours étant  $y$ ) qui n’est pas un de ses descendants dans l’arbre primaire. Ce message provoque l’échange entre l’interface d’entrée et celle de sortie sur tous les nœuds du chemin primaire entre  $x$  et  $y$  et l’installation d’états de routage multicast sur tous les nœuds du chemin de secours (entre  $y$  et  $z$ ). Pour l’exemple de la figure 5 (b), le nœud  $M3$  enverra le message *RECONFIG* au nœud  $C$ , ce qui provoque l’installation d’un nouvel état d’acheminement multicast sur le nœud  $C$  pour permettre de servir de nouveau le nœud  $M3$  (figure 5 (b)). La situation des configurations après la récupération de la panne du lien ( $M3, B$ ) de l’exemple de la figure 4 (a) est illustrée sur la figure 5 (b).

Formellement, les nœuds du réseau exécutent les deux algorithmes 2 et 3 pour rétablir la connexion multicast.

## 4 SIMULATION

Afin d’évaluer la qualité de la protection remontante, nous l’avons comparée à la technique de protection par forêt duale en utilisant le taux de protection comme

métrique. Nous rappelons que le taux de protection mesure la résistance aux pannes et est défini comme le ratio entre le nombre de cas où la technique de protection utilisée réussit à rétablir complètement la communication multicast sur le nombre de cas total.

Pour mieux situer la capacité de la technique de protection remontante à contourner les pannes, nous avons aussi comparé son taux de protection à celui d’une technique de protection idéale.

### 4.1 Modèle de simulation

Le modèle de simulation utilisé ici est le même que celui de [Saidi, 2006]. Dans ce modèle, tous les liens sont bidirectionnels et de coût égal à 1. Les graphes utilisés sont connectés et aléatoirement générés selon l’approche de Waxman [Waxman, 1988]. Les groupes multicast sont aussi aléatoirement choisis parmi les nœuds du graphe.

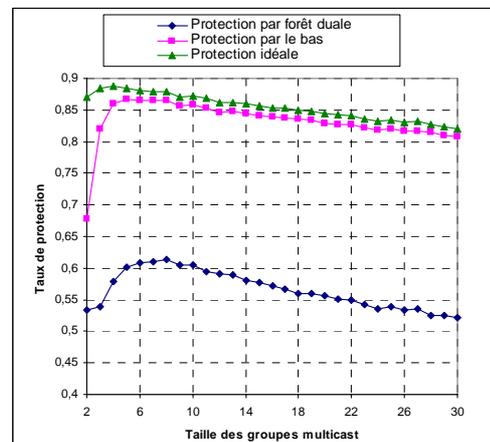
Deux types de pannes ont été considérés : pannes de liens et pannes de nœuds. Les liens tombant en panne sont sélectionnés aléatoirement parmi les liens de l’arbre primaire et les nœuds tombant en panne sont choisis aléatoirement parmi les nœuds internes de l’arbre primaire. Pour chaque panne, les trois techniques de protection sont appliquées afin de déterminer les taux de protection correspondant à chacune d’elles.

Nous ne présentons ici que les résultats obtenus pour le cas de topologies faiblement maillées où la différence entre la protection par forêt duale et la protection remontante est très importante.

### 4.2 Comparaison et analyse

Les deux figures 6 et 7 illustrent les taux de protection des trois techniques de protection (remontante, par forêt duale et idéale) pour les cas de pannes des liens et des nœuds. Concernant la protection remontante, nous précisons ici que le nœud source n’est pas utilisé pour protéger la communication multicast (le couple contenant le nœud source est supprimé de la liste *liens\_arbre\_primaire* de l’algorithme de construction des routes de secours).

Nous remarquons sur les figures 6 et 7 que les taux de protection de la technique de protection idéale est légèrement meilleurs que ceux correspondant à la



**Fig 6. Taux de protection des liens**

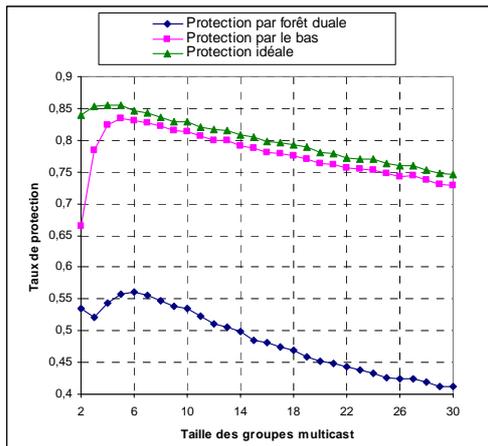


Fig 7. Taux de protection des nœuds

technique de protection remontante qui sont eux-mêmes très supérieurs à ceux de la technique de protection par forêt duale.

Dans le cas de la protection remontante, les nœuds se protègent en utilisant des routes de secours interconnectant (presque) n'importe quels nœuds de l'arbre primaire (avec une priorité pour les nœuds proches des feuilles de l'arbre primaire pour diminuer du coût de la structure de secours) alors que seules les routes interconnectant les nœuds feuilles (sans utilisation de liens ou de nœuds internes de l'arbre primaire) sont utilisées dans le cas de la protection par forêt duale. Cela explique la différence entre les deux taux correspondant à ces deux techniques de protection qui est élevée dans le cas de topologies faiblement maillées où la probabilité d'existence des routes de secours interconnectant deux nœuds feuilles de l'arbre primaire sans utilisation d'un autre nœud ou lien de cet arbre est petite. Concernant la différence très petite entre le taux de protection de la technique de protection remontante et le taux idéal, cela s'explique par l'exploration de presque toutes les routes de secours par la protection remontante. En effet, seules les routes interconnectant deux nœuds de l'arbre primaire ayant un même *numLoop* sont rejetées vu que ces routes n'améliorent pas la protection en général (puisque'il existe, dans la plupart des cas, une boucle passant par les deux nœuds).

Pour les groupes multicast de taille 2 (ou 3), la technique de protection remontante essaye de déterminer des chemins de secours reliant le nœud le plus proche de la destination au nœud source. Ces routes doivent avoir la source comme extrémité, ce qui affaiblit son taux de protection en le comparant à celui d'une technique de protection idéale qui explore toutes les routes de secours possibles. Par contre, le taux de la technique de protection par forêt duale est toujours inférieur aux autres taux car cette technique de protection n'utilise que les routes reliant directement la source à la destination et qui sont disjointes des routes primaires.

## 5 CONCLUSION

Dans cet article, nous avons donné un bref aperçu des techniques de protection proactive multicast. Nous les avons classées en des techniques de protection inspirées

de l'unicast et en des techniques de protection spécifiques au multicast.

Les techniques de protection inspirées de l'unicast présentent divers inconvénients tels que le coût élevé en termes de bande passante des routes de secours et le risque de formation des boucles dans le réseau après la récupération.

Bien que les techniques spécifiques au multicast (protection par forêt duale et protection par arbre redondant) permettent de remédier aux inconvénients précédents, leur applicabilité reste limitée aux topologies fortement maillées où les taux de protection qu'elles fournissent sont certes inférieurs à l'idéal mais assez élevés pour être acceptables.

Afin d'améliorer le taux de protection dans le cas de topologies faiblement maillées, nous avons proposé la technique de protection remontante qui appartient à la seconde classe et qui protège l'arbre multicast à partir du bas i.e. à partir des nœuds feuilles de l'arbre primaire jusqu'au nœud source. Ainsi, les branches de l'arbre primaire seront reliées par des routes de secours permettant le rétablissement de la communication multicast dans le cas d'une panne.

Les simulations montrent que la technique de protection remontante présente un taux de protection très proche de l'idéal et largement supérieur à celui de la technique de protection par forêt duale, ce qui la rend très prometteuse.

## BIBLIOGRAPHIE

- [Fei, 2000] Fei, A., Cui, J., Gerla, M., Cavendish, D.: "A "Dual-Tree" Scheme for Fault-Tolerant Multicast". In Proceedings of IEEE ICC 2001, June 2000.
- [Medard, 1999] Medard, M., Finn, S., Barry, R., Gallager, R.: "Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs". IEEE/ACM Transactions on Networking, vol.7(5), pages 641-652, October 1999.
- [Meyer, 2004] Meyer, P., Van Den Bosch, S., Degrande, N.: "High Availability in MPLS-Based Networks". Alcatel Telecommunication Review, 4th Quarter 2004.
- [Murakami, 1998] Murakami, K., Kim, H. S.: "Optimal capacity and flow assignment for self-healing ATM networks based on line and end-to-end restoration". In IEEE/ACM Transactions on Networking, vol.6(2), pp.207-221, April 1998.
- [Pan, 2005] Pan, P., Swallow, G., Atlas, A.: "Fast Reroute Extensions to RSVP-TE for LSP Tunnels". RFC 4090, May 2005.
- [Ramamurthy, 1999] Ramamurthy, S., Mukherjee, B.: "Survivable WDM Mesh Networks". Part II - Restoration, pp 2023-2030, IEEE ICC 1999.
- [Saidi, 2006] Saidi, M. Y., Cousin, B., Molnar, M.: "Improved Dual-Forest for Multicast Protection". EuroNGI Conference on Next Generation Internet Design and Engineering, Valencia - SPAIN, April 2006.
- [Waxman, 1988] Waxman, B. M.: "Routing of multipoint connections". IEEE Journal Selected. Areas Communications, vol.6(9), Pages 1617-1622, December 1988.